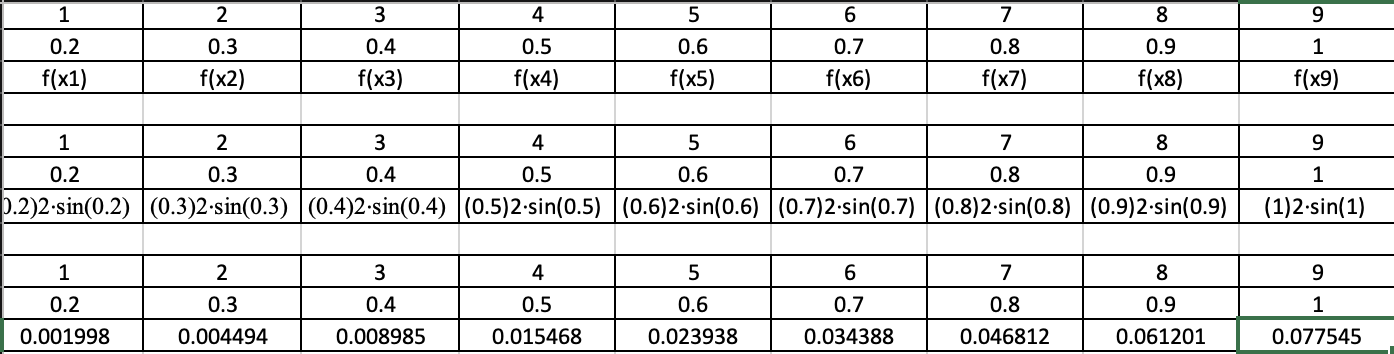
Іванов Кирил

ФІТ 2-8

Варіант 8

𝑓(𝑥) = 𝑥2 ⋅ 𝑠𝑖𝑛(𝑥)



import numpy as np

from scipy.optimize import least\_squares

import matplotlib.pyplot as plt

def fun(a, x, y):

return np.polyval(a, x) - y

x = np.array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])

y = np.array([0.000499, 0.001998, 0.004494, 0.008985, 0.015468, 0.023938, 0.034388, 0.046812, 0.061201,0.077545 ])

a0 = np.array([1, 1])

res\_lsq = least\_squares(fun, x0=a0, args=(x, y))

print("a0 = %.2f, a1 = %.2f" % tuple(res\_lsq.x))

x\_p = np.linspace(min(x), max(x), 20)

y\_p = np.polyval(res\_lsq.x, x\_p)

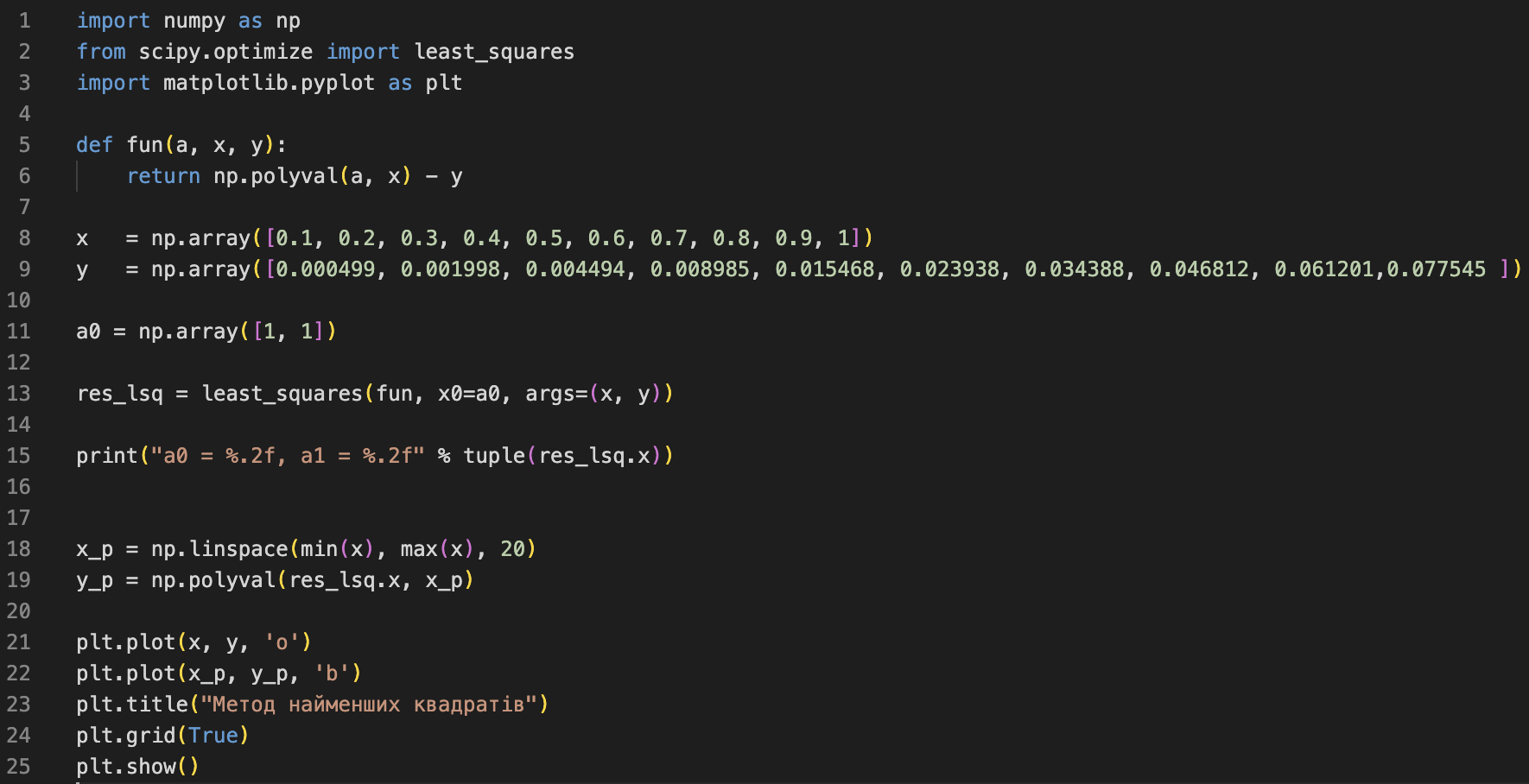
plt.plot(x, y, 'o')

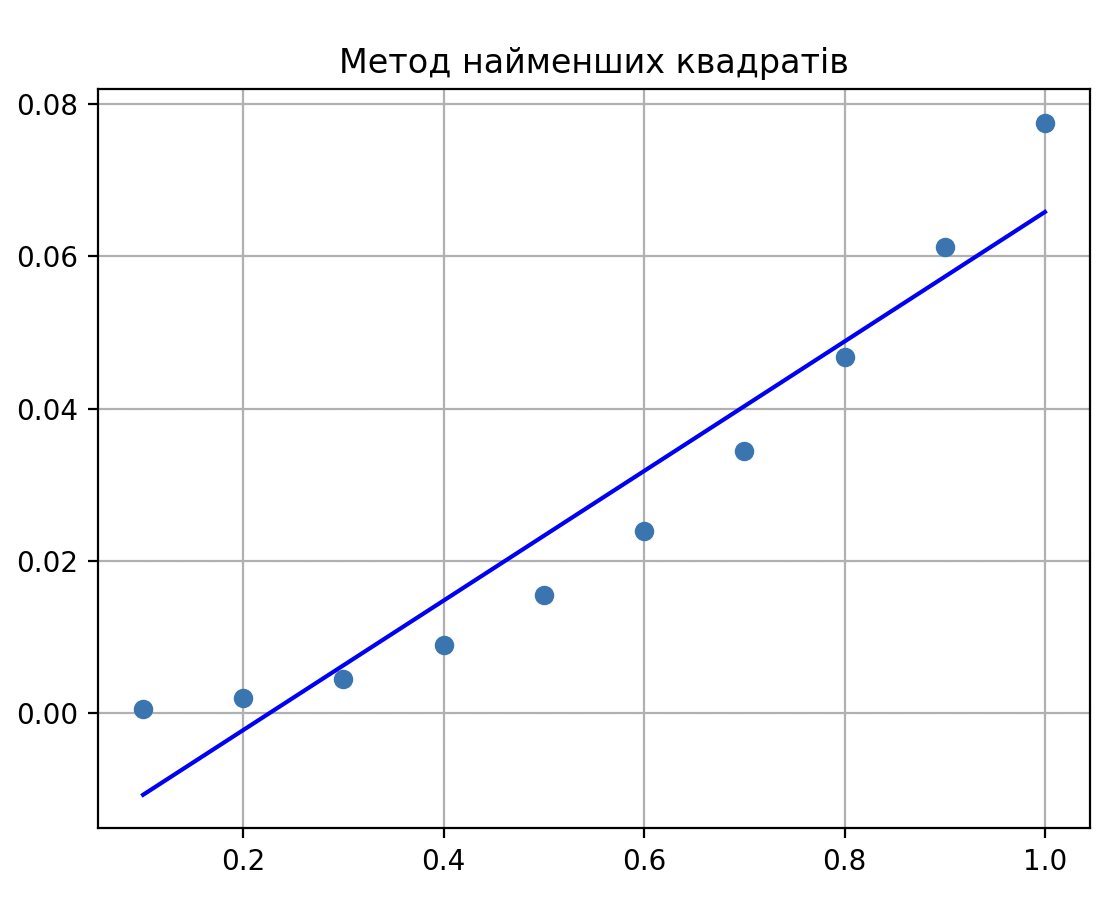
plt.plot(x\_p, y\_p, 'b')

plt.title("Метод найменших квадратів")

plt.grid(True)

plt.show()







import numpy as np

from scipy.optimize import least\_squares

import matplotlib.pyplot as plt

def fun(a, x, y):

return a[0] + a[1] \* x + a[2] \* x\*\*2 -y

x = np.array([0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1])

y = np.array([0.000499, 0.001998, 0.004494, 0.008985, 0.015468, 0.023938, 0.034388, 0.046812, 0.061201,0.077545 ])

a0 = np.array([1, 1, 1])

res\_lsq = least\_squares(fun,x0 = a0, args = (x, y))

print("a0 = %.2f, a1 = %.2f, a2 = %.2f"% tuple(res\_lsq.x))

f = lambda x: sum([u \* v for u, v in zip(res\_lsq.x, [1, x, x\*\*2])])

x\_p = np.linspace(min(x), max(x), 20)

y\_p = f(x\_p)

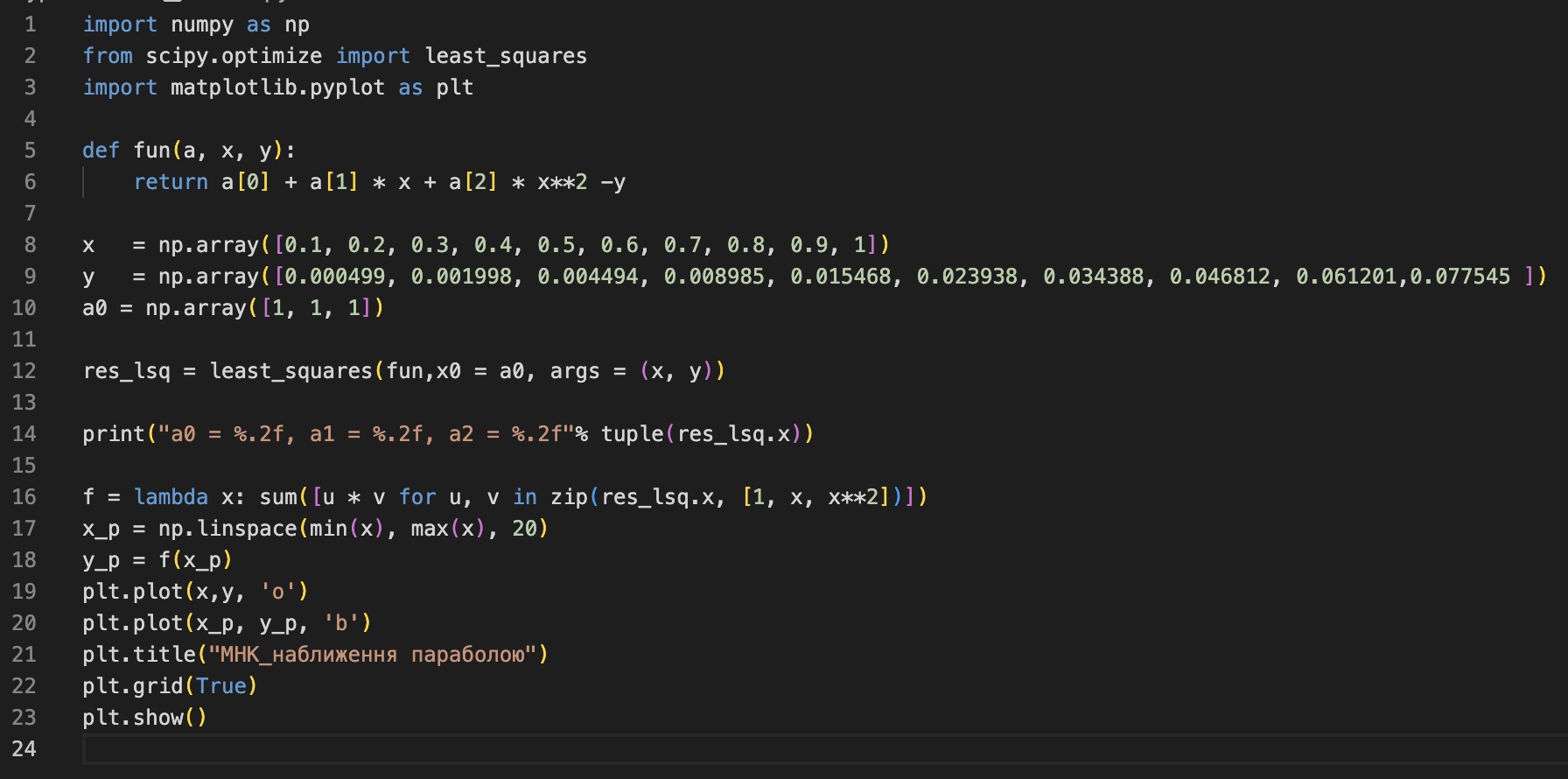
plt.plot(x,y, 'o')

plt.plot(x\_p, y\_p, 'b')

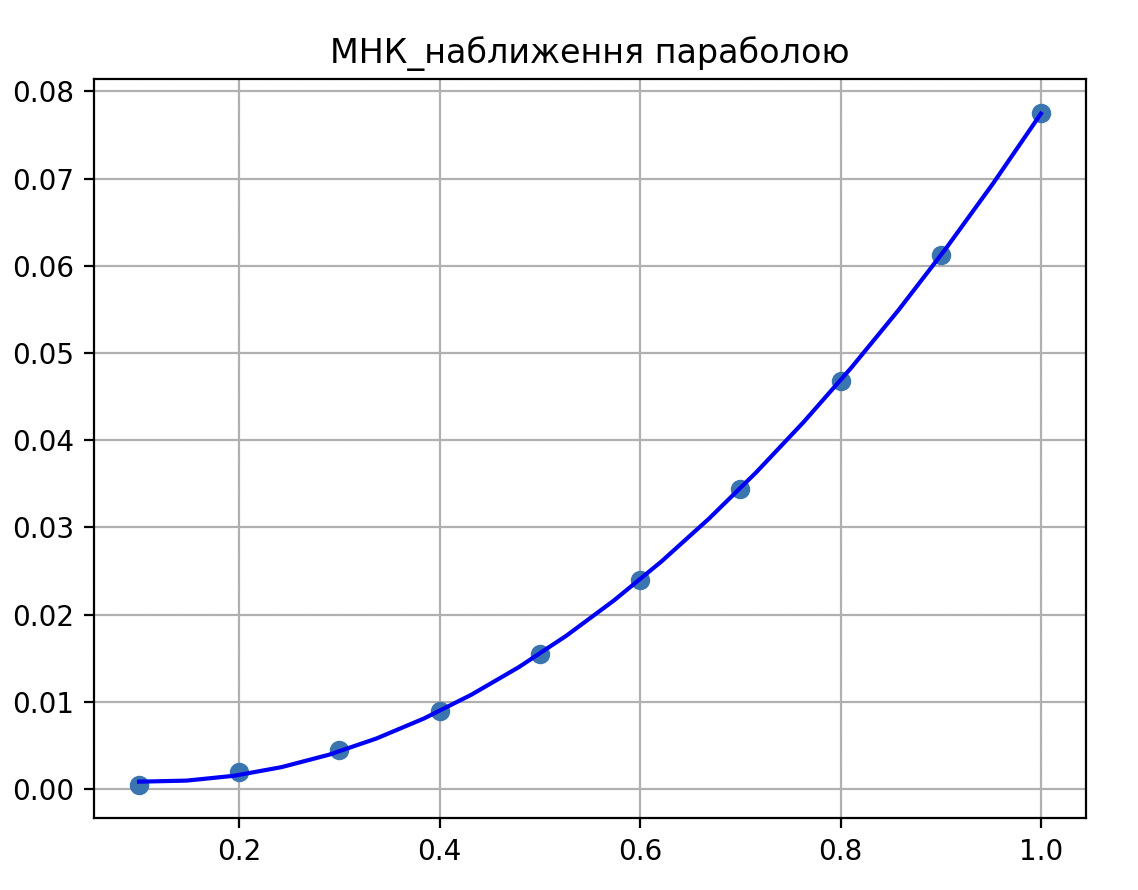
plt.title("МНК\_наближення параболою")

plt.grid(True)

plt.show()







import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import least\_squares

def func(x):

return x\*\*2 \* np.sin(x)

x = np.array([i \* 0.1 for i in range(1, 11)])

y = np.array([func(xi) for xi in x])

print('x=', x)

print('y=', y)

def fun(a, x, y):

return a[0] \* x\*\*2 \* np.sin(x) - y

a0 = np.array([1])

res\_lsq = least\_squares(fun, x0=a0, args=(x, y))

print("a0 = %.2f" % res\_lsq.x[0])

f = lambda x: res\_lsq.x[0] \* x\*\*2 \* np.sin(x)

x\_p = np.linspace(min(x), max(x), 100)

y\_p = f(x\_p)

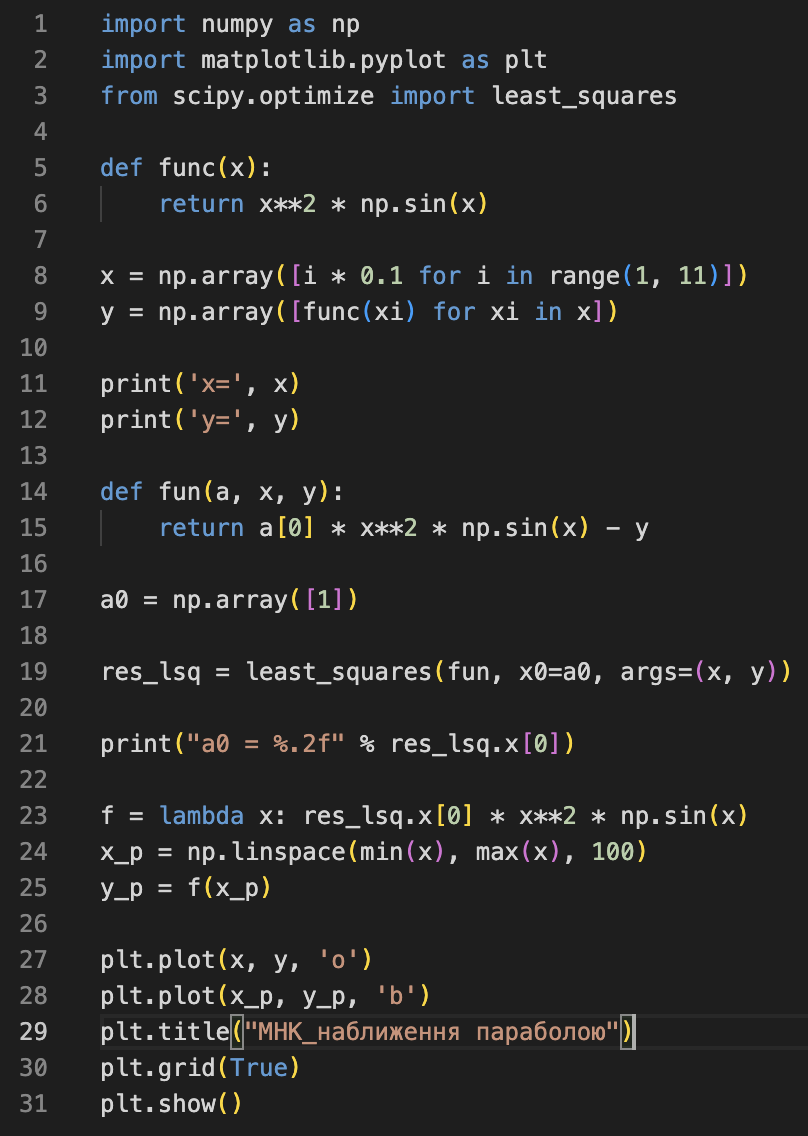
plt.plot(x, y, 'o')

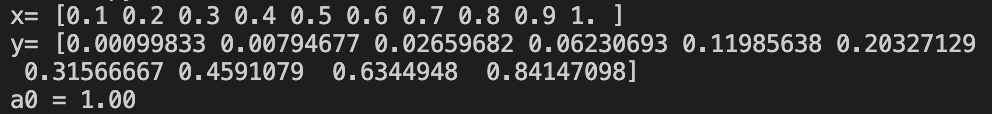
plt.plot(x\_p, y\_p, 'b')

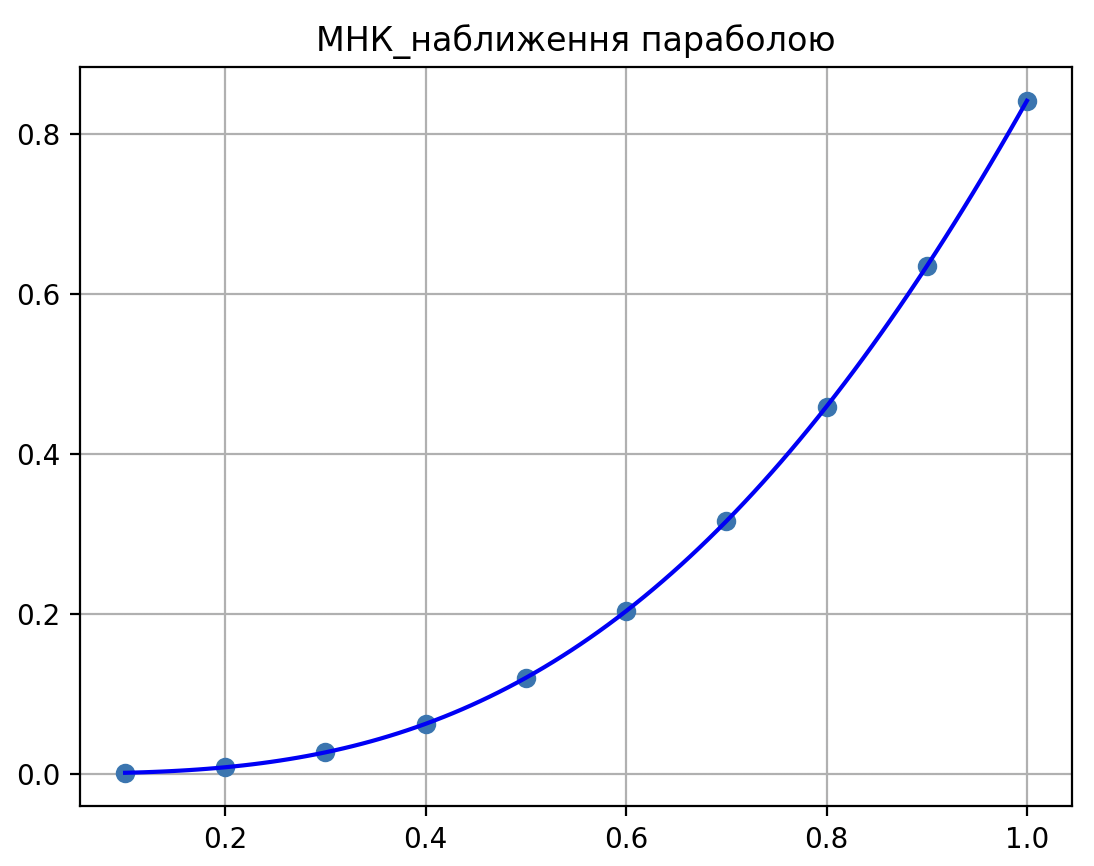
plt.title("МНК\_наближення параболою")

plt.grid(True)

plt.show()







import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import least\_squares

def func(x):

return x\*\*2 \* np.sin(x)

x = np.array([i \* 0.1 for i in range(1, 11)])

y = np.array([func(xi) for xi in x])

print('x=', x)

print('y=', y)

def fun(a, x, y):

return a[0] \* x\*\*2 \* np.sin(x) - y

a0 = np.array([1])

res\_lsq = least\_squares(fun, x0=a0, args=(x, y))

print("a0 = %.2f" % res\_lsq.x[0])

f = lambda x: res\_lsq.x[0] \* x\*\*2 \* np.sin(x)

x\_p = np.linspace(min(x), max(x), 100)

y\_p = f(x\_p)

plt.plot(x, y, 'o')

plt.plot(x\_p, y\_p, 'b')

plt.title("МНК\_наближення прямою")

plt.grid(True)

plt.show()

